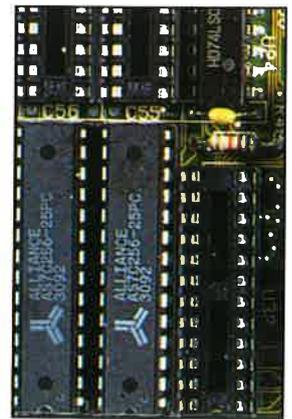
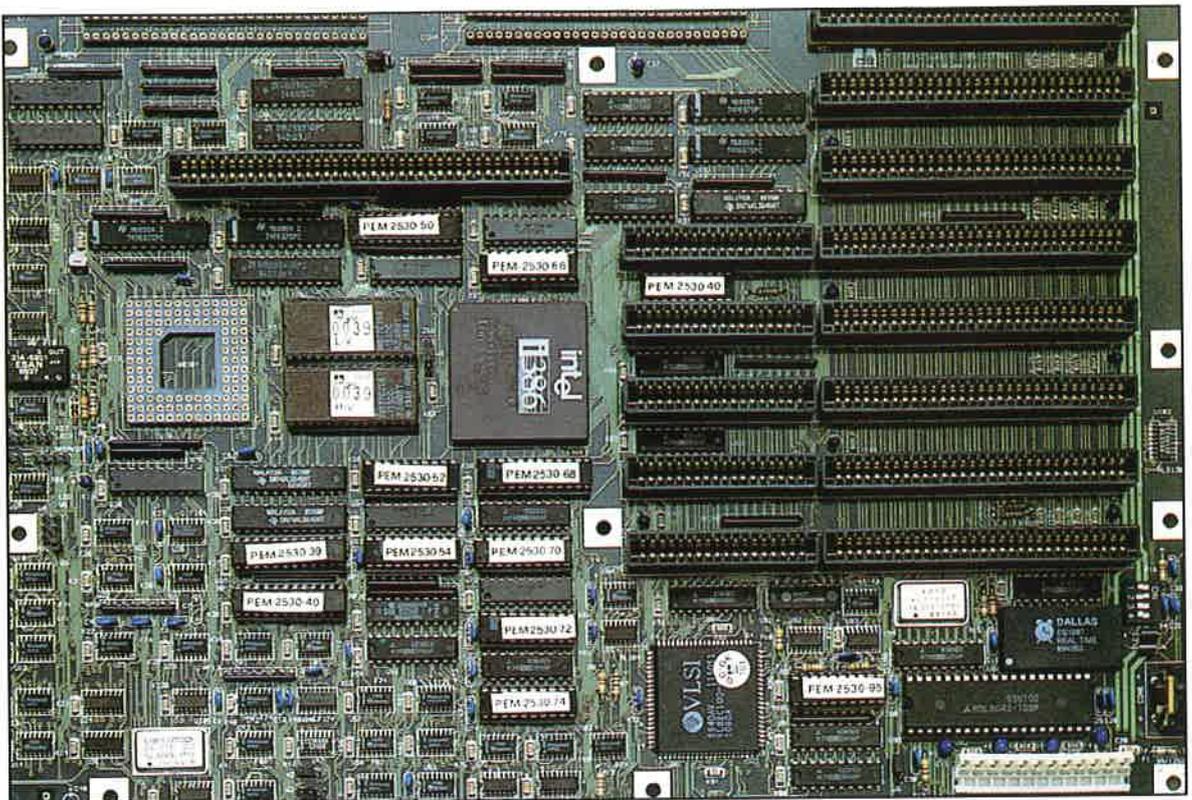
La scheda madre del 386

Il microprocessore 386 preso singolarmente non è in grado di svolgere alcuna funzione, ma deve essere inserito in un contesto hardware imperniato su di esso, che è costituito dalla scheda madre. Questa è stata progettata per ottenere un sistema potente ed efficiente, tenendo presente tutte le caratteristiche funzionali di cui è dotato il microprocessore 386.

In commercio si possono trovare diversi modelli di schede madri 386, in funzione della velocità di lavoro del microprocessore (16, 20, 24, 33, 40 MHz), della dimensione della memoria cache (64, 128 Kbyte) e del tipo di microprocessore (DX, SX, SL).

La scheda madre 386 rappresenta la configurazione minima di elementi

In commercio si possono trovare diversi modelli di schede madri 386

L'insieme della scheda madre e del sistema operativo costituiscono un sistema informatico conosciuto come PC386

Tipica scheda madre 386

circuitali che, raggruppati assieme, formano una entità in grado di eseguire calcoli programmati. Per ottenere un dispositivo completo che soddisfi determinate esigenze, a questo insieme di circuiti, essenziali per il funzionamento di un sistema 386, è necessario aggiungere altri elementi (scheda controller per i dischi, porte seriali e parallele, scheda grafica, modem, schede di acquisizione dati, ecc.).

Disposizione convenzionale dei diversi componenti hardware presenti sulla scheda 386

Su questa scheda viene caricato il sistema operativo, che consente l'esecuzione delle diverse applicazioni scelte dall'utente.

La scheda 386 può supportare diversi tipi di sistemi operativi, quali l'MS-DOS e il DR-DOS per applicazioni monoutente, e l'UNIX per applicazioni multiutenza.

L'insieme della scheda madre e del sistema operativo costituiscono un sistema informatico conosciuto come PC386.

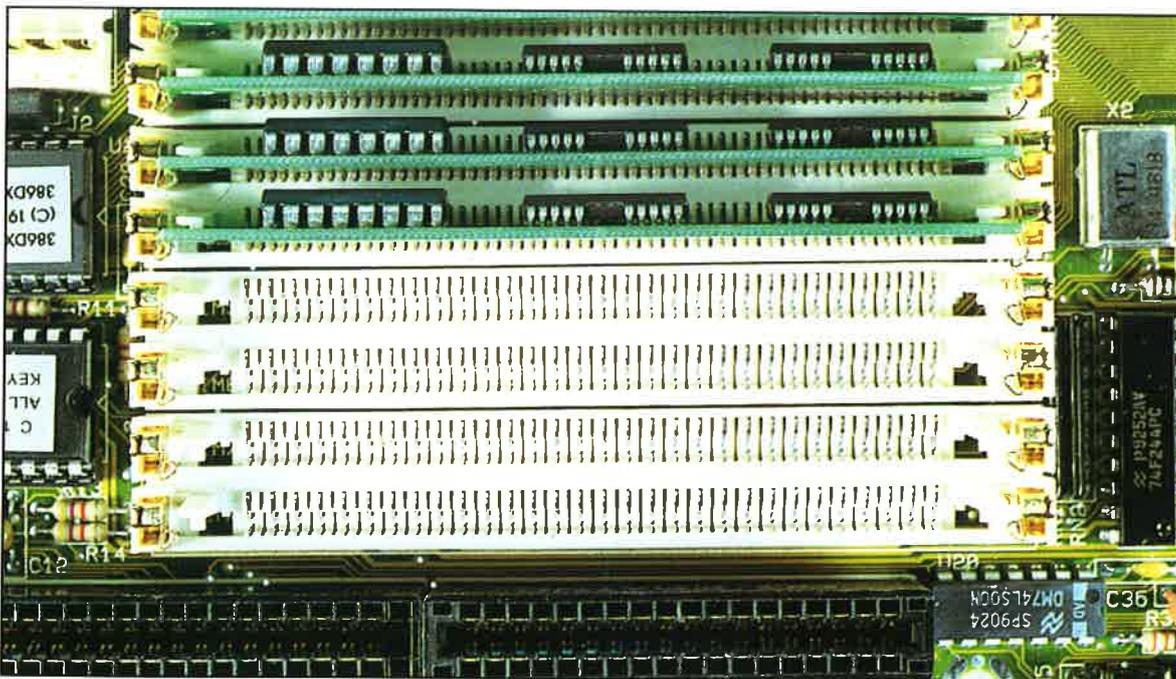
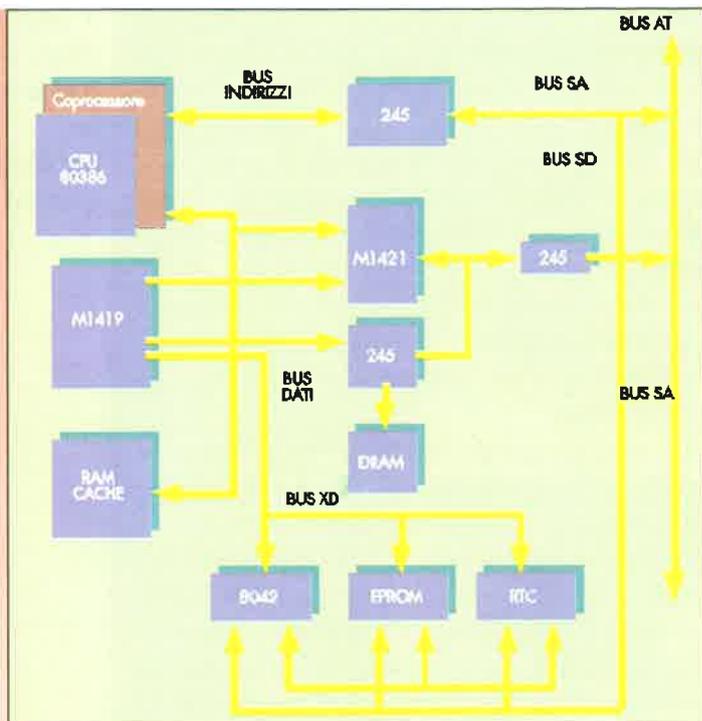
Struttura della scheda madre

Per la costruzione della scheda madre 386 viene generalmente impiegata l'architettura *ISA* (*Architettura Standard dell'Industria*), che è la più utilizzata per la costruzione dei personal computer, anche se attualmente si sta imponendo l'architettura *EISA* (*Architettura Standard dell'Industria Avanzata*).

La differenza principale che esiste tra questi due tipi di architettura è dovuta alle diverse caratteri-

stiche del bus di ingresso/uscita.
La scheda madre 386, costruita con architettura ISA, è in grado di gestire una serie di segnali che possono essere così raggruppati:
segnale di clock,
segnali relativi agli indirizzi,
segnali di controllo dei cicli,
segnali di controllo degli interrupt,
segnali per il DMA,
segnali di alimentazione.
- Il segnale di clock è quello che determina il ritmo di lavoro del sistema, e viene generato da un oscillatore al quarzo che fornisce una frequenza doppia rispetto a quella utilizzata dal microprocessore; infatti, per operare in modo corretto questo valore di frequenza viene internamente diviso per due. Oltre alla frequenza propria di sistema, sono necessarie anche altre frequenze di temporizzazione per l'esecuzione di certi compiti particolari, quali la richiesta di refresh della memoria e la funzione di orologio/calendario.
- I transcettori degli indirizzi sono suddivisi in due gruppi: gli *indirizzi di sistema*, costituiti dalle linee SA0-SA19 e gli *indirizzi periferici*, formati dalle linee LA16-LA23.
Un segnale conosciuto con il nome di BALE, ha il compito di convalidare gli indirizzi del bus.
Sul fronte di discesa del segnale BALE vengono confermati gli indirizzi di sistema, per cui è possibile indirizzare fino ad 1 Megaottetto di memoria.
Gli indirizzi LA16-LA23, quando il segnale di conferma (BALE) fornisce il relativo consenso, possono invece indirizzare 16 Megaottetti di memoria.
- Il bus dati è costituito dalle linee SD, che vengono generalmente gestite con un particolare sistema multiplexing in cui i dati sono normalmente costituiti da ottetti.
- Un altro gruppo molto importante di segnali è quello per il controllo dei cicli: di lettura e scrittura, di ingresso/uscita e di trasferimento della memoria. Per i cicli di lettura e scrittura sono utilizzati i segnali MRDC# (per la lettura) e MWTC# (per la scrittura). Il controllo dei cicli di ingresso e di uscita sugli slot di espansione è assegnato rispettivamente ai segnali IORC# e IOWC#. Il trasferimento dei dati tra la memoria e i dispositivi di ingresso/uscita è pilotato dai segnali M16# e IO#.
Quando si accede a dispositivi periferici lenti è necessario prolungare il tempo di accesso tramite dei cicli di attesa, durante i quali il segnale CHRDY# è impostato a livello basso.
Al contrario, se i dispositivi periferici sono molto

Il segnale di clock è generato da un oscillatore al quarzo

Banchi di memoria RAM della scheda 386, implementati tramite elementi di memoria conosciuti con il nome di SIMM

Schema a blocchi di una scheda madre 386

veloci è necessario abbreviare il tempo di accesso, per cui il segnale NOWS deve essere impostato a 0.

- Gli interrupt vengono gestiti da un controllore degli interrupt; questo circuito, a fronte della richiesta di interrupt di una determinata periferica, valuta la priorità della stessa determinando il peso che gli è stato dato con il livello IRQ di interruzione, e la invia di conseguenza al microprocessore. Ciò significa che quando arrivano più richieste di interrupt simultanee, il controller invia per prima al microprocessore quella di maggior peso, e successivamente le altre in funzione della loro priorità.

La funzione del DMA consente di trasferire una informazione ad alta velocità senza l'intervento della CPU

Gli interrupt controllati da questo dispositivo sono:
- *NM1, controllo di parità o controllo del canale di ingresso/uscita,*
- *IRQ0, temporizzatore di sistema,*
- *IRQ1, tastiera,*
- *IRQ2, controllo degli interrupt da IRQ8 a IRQ15,*
- *IRQ3, porta seriale 2,*
- *IRQ4, porta seriale 1,*
- *IRQ5, porta parallela 2,*
- *IRQ6, floppy drive,*
- *IRQ7, porta parallela 1,*
- *IRQ8, clock in tempo reale,*
- *IRQ9, IRQ10, IRQ11, IRQ12, interrupt riservati,*
- *IRQ13, coprocessore matematico 80387,*
- *IRQ15, interrupt riservato,*

- I canali DMA generano indirizzamenti di memoria e segnali di controllo. Questi segnali sono necessari per il trasferimento diretto delle informazioni tra un dispositivo periferico e la memoria. La funzione DMA consente il trasferimento di una informazione ad alta velocità, con una necessità minima di intervento da parte della CPU. Generalmente in una scheda 386 sono presenti due controller DMA, che permettono la gestione di 4 canali per il trasferimento dei dati alle periferiche a 8 bit (DMA1), e di tre canali per il trasferimento dei dati alle periferiche a 16 bit (DMA2). Il controllo della lunghezza del periodo del ciclo è normalmente già disponibile all'interno del circuito integrato per il DMA, e consente di gestire in modo indipendente i cicli a 8 e 16 bit. Questa funzione viene eseguita attraverso i registri di tipo programmabile, che possono emettere segnali di comando o inserire cicli di attesa. Ciascun canale DMA è dotato di una coppia di contatori a 16 bit e di un registro per ciascun contatore.

Dettaglio della ROM-BIOS di una scheda madre del PC386

Questi contatori a 16 bit consentono di trasferire blocchi da 65.536 parole.
Il circuito DMA si trova per default in condizione di "attesa", per cui esegue cicli ripetitivi relativi ad un solo stato.
Quando si verifica una richiesta di DMA viene attivata la funzione di accesso diretto alla memoria, che viene inviata al sistema per generare l'indirizzamento di memoria e i segnali di comando necessari per completare il trasferimento dalla memoria all'interfaccia di ingresso/uscita, oppure dall'interfaccia di ingresso/uscita alla memoria, oppure dalla memoria alla memoria.
Per il suo funzionamento la scheda madre ha bisogno di diverse alimentazioni in corrente continua, che vengono prodotte da un alimentatore e trasferite alla scheda stessa tramite due gruppi di cavi e due connettori.
Le alimentazioni necessarie alla scheda madre 386 sono:
+5 Vcc, -5 Vcc, +12 Vcc e -12 Vcc

Il coprocessore matematico 80387

La scheda madre 386 è dotata di uno zoccolo per l'installazione (opzionale) del coprocessore matematico 80387. Questo componente è in grado di risolvere autonomamente le istruzioni aritmetiche, e sviluppare diverse funzioni trigonometriche e non, quali tangenti, seni, coseni e logaritmi.
Per eseguire questi calcoli aritmetici, l'80387 si sostituisce effettivamente all'insieme dei registri e delle istruzioni del 386 con altre istruzioni e dati addizionali.
L'80387 non è sensibile alla modalità di funzionamento del 386, ma reagisce sempre allo stesso modo, sia che il 386 lavori in modalità ad indirizzamento reale, in modalità protetta, o in modalità virtuale 8086.
Gli accessi alla memoria del 387 e i valori che gli vengono inviati sono comunque gestiti dal microprocessore. Tutta la comunicazione tra il microprocessore e il coprocessore è trasparente al software applicativo. Il 386 controlla automaticamente il 387 quando viene eseguita qualsiasi istruzione numerica. La memoria, sia fisica che virtuale, è disponibile per memorizzare le istruzioni e gli operandi dei programmi applicativi che lavorano con il coprocessore.

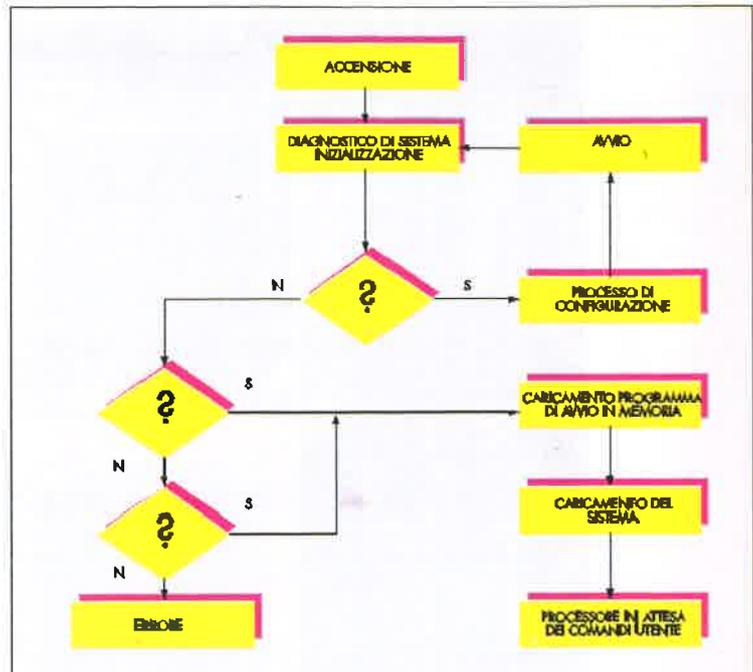
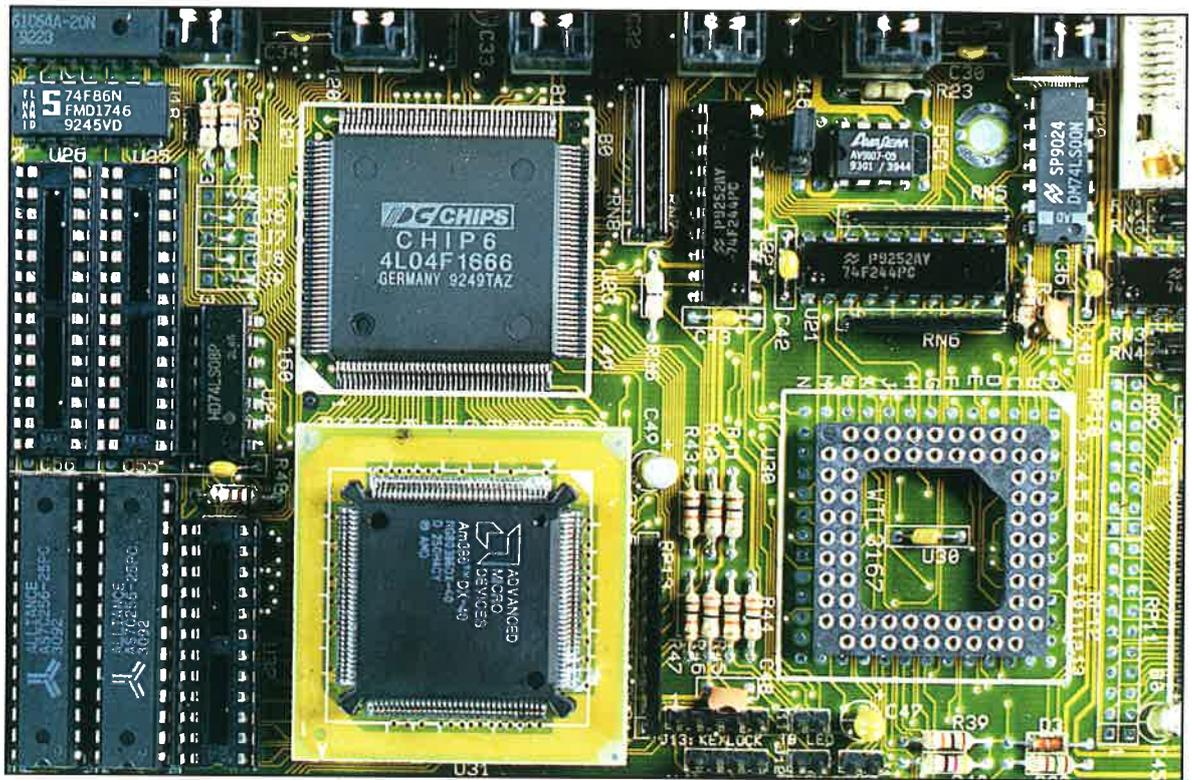



Diagramma di flusso delle fasi di avvio di un PC 386

Il coprocessore non possiede una propria unità per il calcolo e la gestione degli indirizzi, ma ammette tutti i tipi di indirizzamento che è in grado di eseguire la CPU.

Collegamento dell'80387 alla scheda madre

Il coprocessore matematico 80387 sfrutta lo stesso generatore di clock del microprocessore 386, e lavora alla stessa frequenza di clock della CPU in modo sincrono. Funziona come un dispositivo di ingresso/uscita attraverso l'interfaccia di I/O, sfruttando gli indirizzi esadecimali 00F8, 00FA e 00FC.
Il microprocessore 386 invia dalla scheda madre i codici operativi e gli operandi tramite la porta di I/O.
Durante l'esecuzione di una istruzione, il coprocessore può generare degli interrupt, o per meglio dire delle *eccezioni*, che possono essere identificate dalle funzioni:
- *operazione non valida,*
- *operando denormalizzato,*
- *divisione per zero,*
- *overflow,*
- *underflow,*
- *perdita di precisione.*

Le alimentazioni richieste da una scheda madre 386 sono +5 Vcc, –5 Vcc, +12 Vcc e –12 Vcc

A destra, zoccolo per l'inserimento del coprocessore matematico 387

IL CONTROLLER INTEGRATO PER LA MEMORIA CACHE 385

Questo controller agisce da intermediario tra la CPU e la memoria principale, poiché il suo scopo essenziale è fare in modo che la maggior parte della memoria DRAM (*Memoria Dinamica ad Accesso Casuale*) ad alta integrazione, basso costo, e velocità limitata venga identificata dalla CPU come una memoria veloce.
Per ottenere questa condizione viene utilizzata una memoria molto veloce (cache) di piccole dimensioni (da 32 a 128 Kbyte), nella quale vengono memorizzati i dati e le istruzioni utilizzate con maggior frequenza dal 386 durante l'esecuzione di un programma applicativo.
Di conseguenza, se il dato o l'istruzione richiesta non deve essere prelevato dalla memoria principale, ma si trova già presente nella cache, il tempo di accesso si riduce considerevolmente; ciò consente di poter sfruttare una porzione compresa tra l'85% e il 99% della memoria principale come memoria rapida, poiché nelle fasi di accesso il microprocessore non deve eseguire dei cicli di attesa.

Il controller di memoria cache 385 agisce da intermediario tra la CPU e la memoria principale

TIPI DI SCHEDA IN FUNZIONE DEL MICROPROCESSORE

In commercio esistono diversi tipi di schede madri 386, che differiscono tra di loro per il modello di microprocessore 386 di cui sono dotate.
I diversi tipi di processori sono:
- *microprocessore 386 DX,*
- *microprocessore 386 SX,*
- *microprocessore 386 SL.*

386 DX

Il 386 DX è il microprocessore più importante della famiglia 386, e i costruttori di personal computer lo considerano il proseguimento logico dei computer 286 verso livelli superiori.
Questo processore è stato sviluppato in modo da renderlo compatibile verso il 286, a causa della grande quantità di software già scritto per quest'ultimo dispositivo.
Il 386 DX può essere associato ad altri due componenti di notevole importanza, il coprocessore matematico 387 e il controller per la memoria cache 385.

La potenza di questo microprocessore è notevole, poiché supera di diverse volte quella del minielaboratore industriale standard "VAX 11/70".

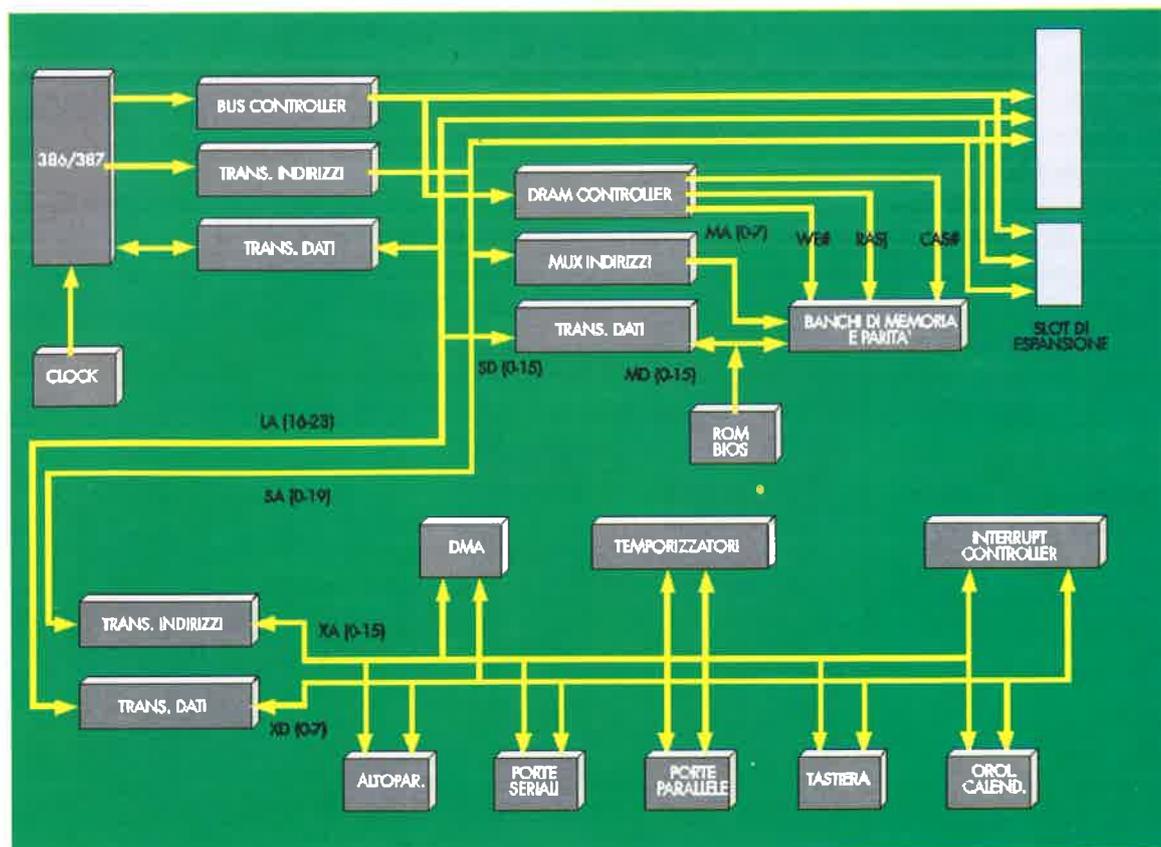
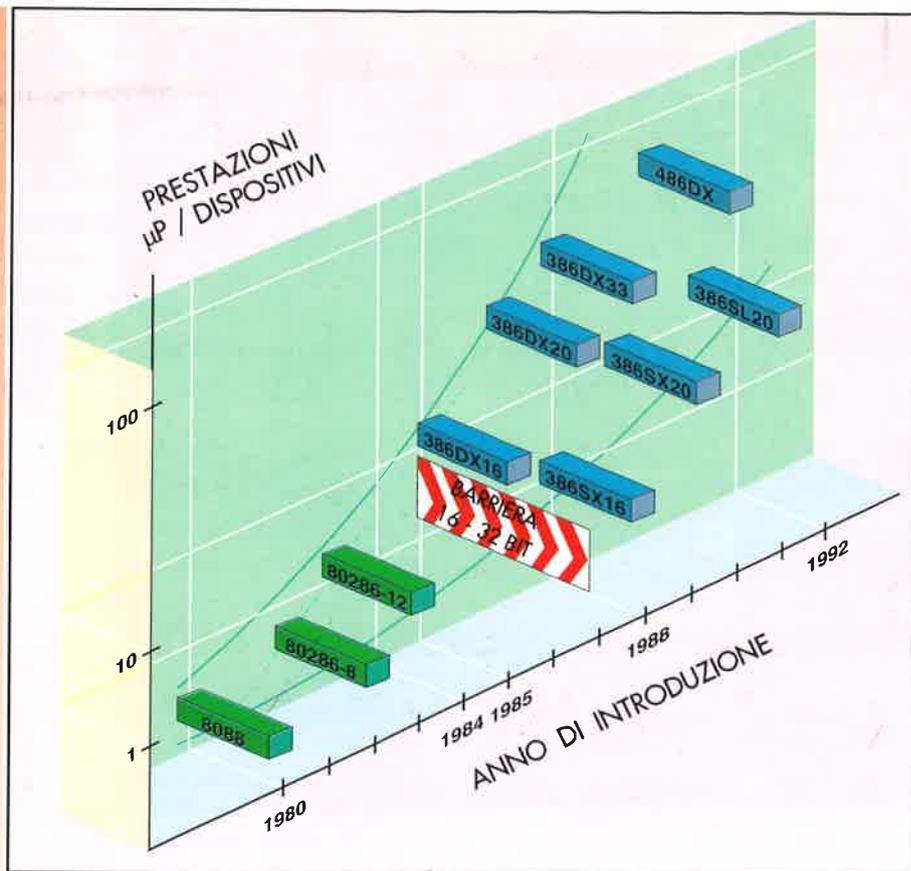
386 SX

La scheda madre 386SX è dotata del microprocessore 386 SX; il suo costo è decisamente inferiore rispetto alla scheda madre 386DX grazie al basso costo del processore stesso e degli elementi circuitali periferici.
Il microprocessore 386 SX sfrutta lo stesso set di istruzioni a 32 bit del 386 DX, per cui è in grado di eseguire il software sviluppato per quest'ultimo processore.
La differenza reale consiste nel fatto che il microprocessore 386 SX comunica con il mondo esterno con un bus a 16 bit, e ciò determina una riduzione delle prestazioni di circa il 10% rispetto al 386 DX.
Uno dei vantaggi che presenta è costituito dal fatto che può utilizzare periferiche e componenti di minor costo, grazie al bus esterno più piccolo rispetto a quello del 386DX; inoltre, anche il contenitore del processore è di tipo plastico, per cui il suo costo risulta decisamente inferiore rispetto a quello ceramico.
I computer portatili, o lap top, sono spesso dotati di questo tipo di scheda, poiché grazie alle sue notevoli prestazioni può far comodo a quelle persone che hanno la necessità di avere a disposizione ovunque un computer pratico e potente.

386 SL

Un altro tipo di scheda è quella che monta il microprocessore modello 386 SL, dotato delle stesse prestazioni degli altri 386 ma con un basso assorbimento; questa condizione si ottiene grazie ad una gestione molto particolare del consumo di energia e ad un sistema costituito da diversi componenti supportati dall'integrato 82360SL.
Queste caratteristiche permettono una riduzione del peso complessivo, e una durata delle batterie

Il 386 DX è il microprocessore più importante della famiglia 386

Schema a blocchi di un 386 costruito con architettura ISA

Evoluzione dei personal computer

di alimentazione superiore di circa il 50 % rispetto ai portatili dotati di altri microprocessori della serie 386.

Quest'ultima condizione si ottiene perché il processore 386SL è stato progettato per poter funzionare in modo "statico"; ciò gli consente di bloccare il clock senza perdere i dati memorizzati nei suoi registri interni.

Inoltre, permette una programmazione sufficientemente flessibile delle frequenze di clock per i diversi dispositivi, ed il mantenimento dello stato della periferica quando viene bloccato il clock di sistema.

La scheda madre 386SL è costituita da cinque soli circuiti integrati, se si esclude la memoria; se si considera anche questa il numero degli integrati aumenta da cinque a nove. Da ciò si può facilmente dedurre che questa scheda ha dimensioni notevolmente ridotte, se confrontate con quelle precedenti; questa caratteristica la rende particolarmente interessante per coloro che devono spostarsi frequentemente o il cui posto di lavoro non è il classico ufficio, per cui viene molto utilizzata nei computer note book.

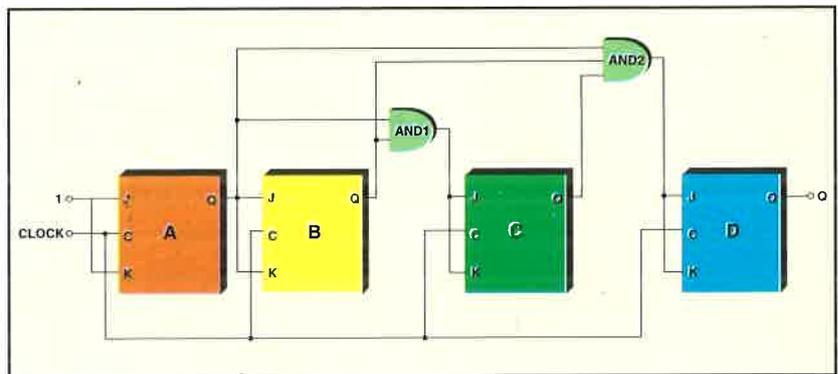
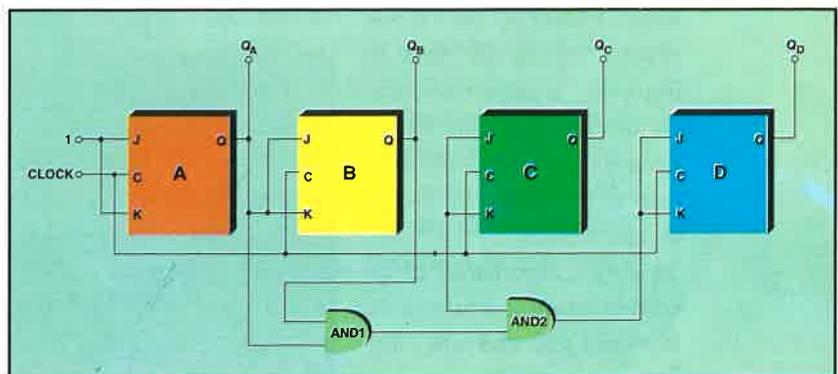
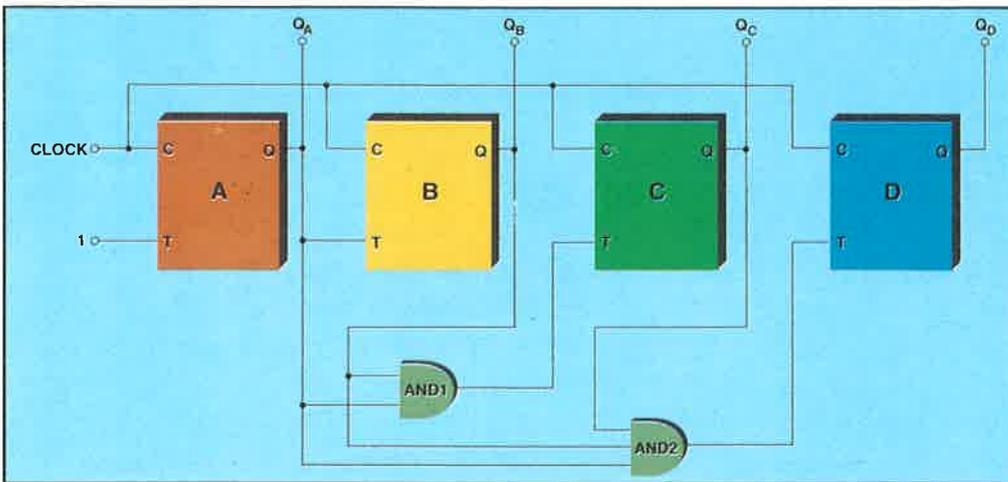
Il 386SL introduce una nuova modalità di funzionamento, conosciuta con il nome di *interrupt per la gestione del sistema*, la cui principale funzione è quella di interrompere tutte le attività e le operazioni in uno stato determinato, lasciando l'apparecchiatura a riposo con tutti i sottosistemi disattivati, ma consentendo però una successiva ripresa del lavoro ed il recupero di tutte le situazioni nello stato in cui erano state lasciate, senza il bisogno di ripetere la procedura di avvio.

In tutte le schede madri 386 è necessario impostare la configurazione di "SETUP" di avvio dell'elaboratore

La batteria mantiene alimentata la memorie di sistema e consente all'orologio/calendario di rimanere attivo anche quando viene scollegata l'alimentazione

I CONTATORI SINCRONI

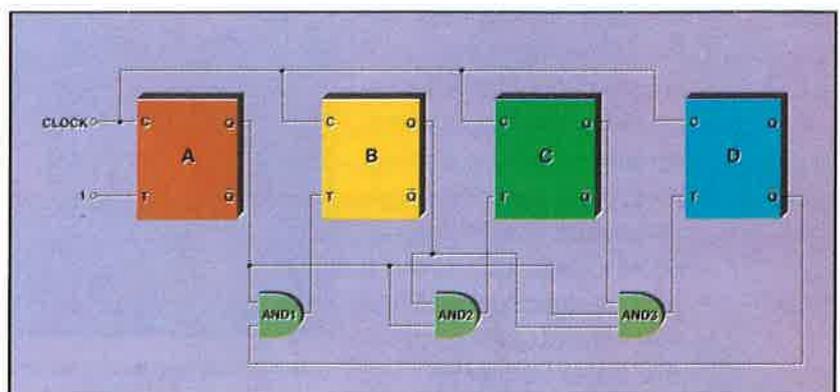
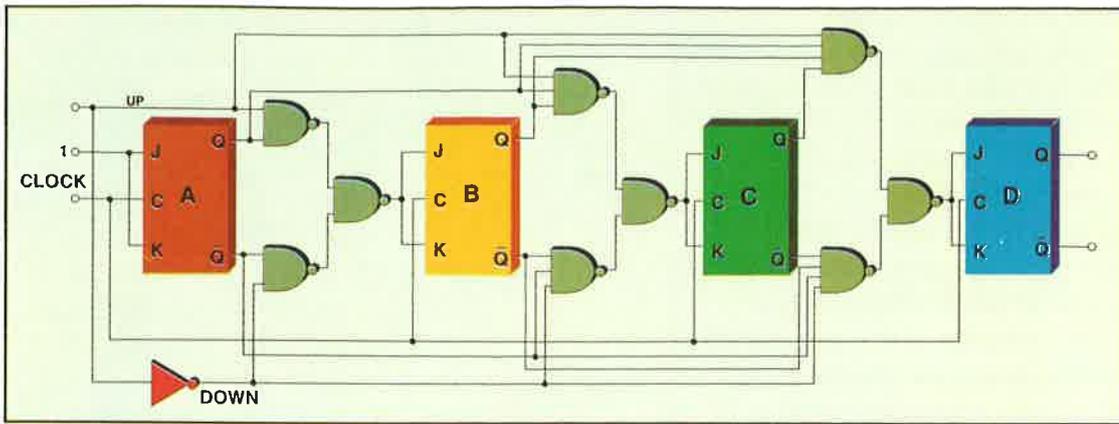
Nei contatori asincroni, esaminati nel capitolo precedente, i bistabili commutavano in modo seriale. Con i contatori sincroni invece, è possibile trasportare i dati sia in modo seriale che parallelo.

Nel capitolo precedente si è visto che i bistabili utilizzati nei contatori asincroni hanno un segnale fisso applicato in modo permanente ai loro ingressi J-K o T, e che la loro commutazione viene abilitata da un segnale di clock. Quest'ultimo proviene dal bistabile precedente per cui, per ottenere la commutazione dell'ultimo bistabile della catena, è necessario che tutti quelli che lo precedono commutino il loro stato. Di conseguenza, se il ritardo di ogni bistabile viene indicato con t, un contatore formato da n bistabili tenderà a cambiare completamente di stato dopo un ritardo totale pari a n x t; questa condizione limita l'impiego di questi contatori alle basse frequenze.

Utilizzando la trasmissione seriale, come avviene in un contatore asincrono, i tempi di propagazione aumentano limitando la frequenza degli impulsi di clock

Contatore sincrono con trasmissione seriale

I bistabili che normalmente vengono utilizzati per la costruzione dei contatori sincroni sono di tipo T

Volendo verificare questa affermazione, si può considerare la situazione seguente. Il tempo di propagazione è il tempo richiesto da un contatore per completare la sua risposta ad un impulso di ingresso. Questo tempo, in un contatore asincrono, è direttamente proporzionale al numero di bistabili che formano il dispositivo, e perciò al numero di stadi che devono essere attivati in successione. In questa condizione infatti, un nuovo impulso fa cambiare di stato solo tutti i flip-flop che hanno già subito una commutazione in precedenza.
Qualsiasi bistabile, infatti, non risponde finché lo stadio che lo precede non ha completato la transizione; di conseguenza, poiché l'impulso di clock si propaga attraverso la catena dei flip-flop, il suo tempo di trasferimento assume un valore finale pari alla somma dei ritardi di propagazione di tutti i bistabili interessati. Se è presente un numero di stadi sufficientemente alto, il tempo di trasferimento può risultare maggiore dell'intervallo di tempo che trascorre tra due diversi impulsi di clock di ingresso; questa condizione rende praticamente impossibile la lettura del contatore tra due impulsi di clock.
Se il funzionamento di un contatore viene trasformato in modo tale che tutti i flip-flop risultino azionati contemporaneamente dall'impulso di clock ingresso, il tempo di propaga-zione può essere notevolmente ridotto. La cadenza della ripetizione è limitata solo dal ritardo di uno qualsiasi dei bistabili e dal tempo di propagazione delle porte di controllo presenti. Normalmente la frequenza massima di funzionamento di un contatore sincrono con quattro bistabili è di 55 MHz, quasi il doppio di quella a cui può lavorare un contatore asincrono.
Un altro vantaggio del contatore sincrono è che sull'uscita i segnali sono sempre validi, poiché tutti i flip-flop cambiano di stato contemporaneamente. Di conseguenza, quando si decodifica un contatore sincrono non è richiesto un impulso di conferma dei dati.
Questo fatto verrà esaminato con maggior dettaglio quando, nei prossimi capitoli, verranno trattati i decodificatori.
I contatori sincroni possono trasmettere i dati presenti ai loro ingressi sia in modo seriale che parallelo.

TRASMISSIONE SERIALE

Come si può osservare nella figura corrispondente, il contatore sincrono è formato da quattro bistabili, tutti di tipo T, ottenuti partendo da flip-flop J-K nei quali i terminali J e K vengono collegati tra di loro. Come già si è visto nei capitoli precedenti, se all'ingresso di un bistabile di tipo T

Contatore sincrono decadico con trasmissione parallela

Contatore up/down con trasmissione parallela, nel quale le porte di controllo consentono di eseguire entrambe le funzioni

è presente lo stato logico 0, quando viene applicato un impulso di clock non si verifica alcun cambio di stato sulle uscite. Se invece J=K=1, l'uscita del flip-flop viene complementata ad ogni impulso.

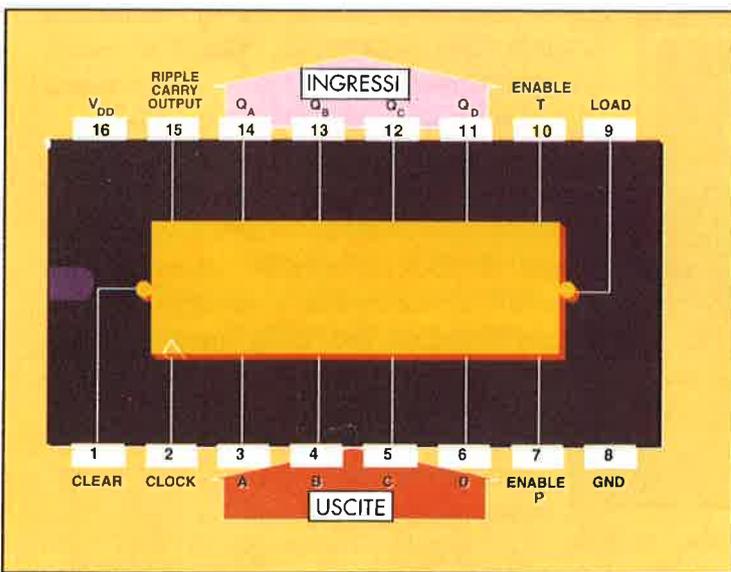
Per ottenere un diagramma dei tempi analogo a quello ottenuto per il contatore asincrono si devono fare le seguenti premesse:

- per fare in modo che Q_A commuti a ciascun impulso di clock il valore dell'ingresso del primo bistabile deve essere 1,
- Q_B può essere complementato da un impulso di clock solo quando Q_A vale 1, per cui l'ingresso del bistabile B è definito da Q_A,
- solo quando Q_A e Q_B assumono valore 1 Q_C può commutare a $/Q_C$, per cui l'ingresso di questo flip-flop è definito dall'equazione $Q_A \times Q_B$.
- proseguendo con questa logica, l'uscita dell'ultimo bistabile viene abilitata alla commutazione solo quando l'ingresso risulta definito dall'equazione combinatoria: $Q_A \times Q_B \times Q_C$.

Di conseguenza, per poter ottenere le suddette condizioni è necessario utilizzare due porte AND, come indicato nella figura. Il tempo minimo che deve intercorrere tra due impulsi di clock successivi è pari all'intervallo richiesto da ciascun J e K per raggiungere lo stato permanente, per cui la frequenza massima degli impulsi di clock deve essere uguale o inferiore all'inverso di questo valore.

TRASMISSIONE PARALLELA

La frequenza massima di funzionamento può essere aumentata utilizzando la trasmissione parallela, nella quale l'impulso di abilitazione di ciascun bistabile proviene da una porta AND a ingressi multipli costituiti dalle uscite dei flip-flop precedenti.

Utilizzando la stessa logica seguita per la descrizione della trasmissione seriale, si può dire che il segnale di ingresso dell'ultimo bistabile proviene da una porta AND a tre ingressi alimentata dalle uscite Q_A, Q_B e Q_C. Il tempo minimo tra due impulsi di clock nella trasmissione di tipo parallelo, che è determinato dalla somma tra il tempo di propagazione di un flip-flop e quello di una porta AND, risulta decisamente inferiore

Schema di collegamento di un contatore sincrono a quattro bit

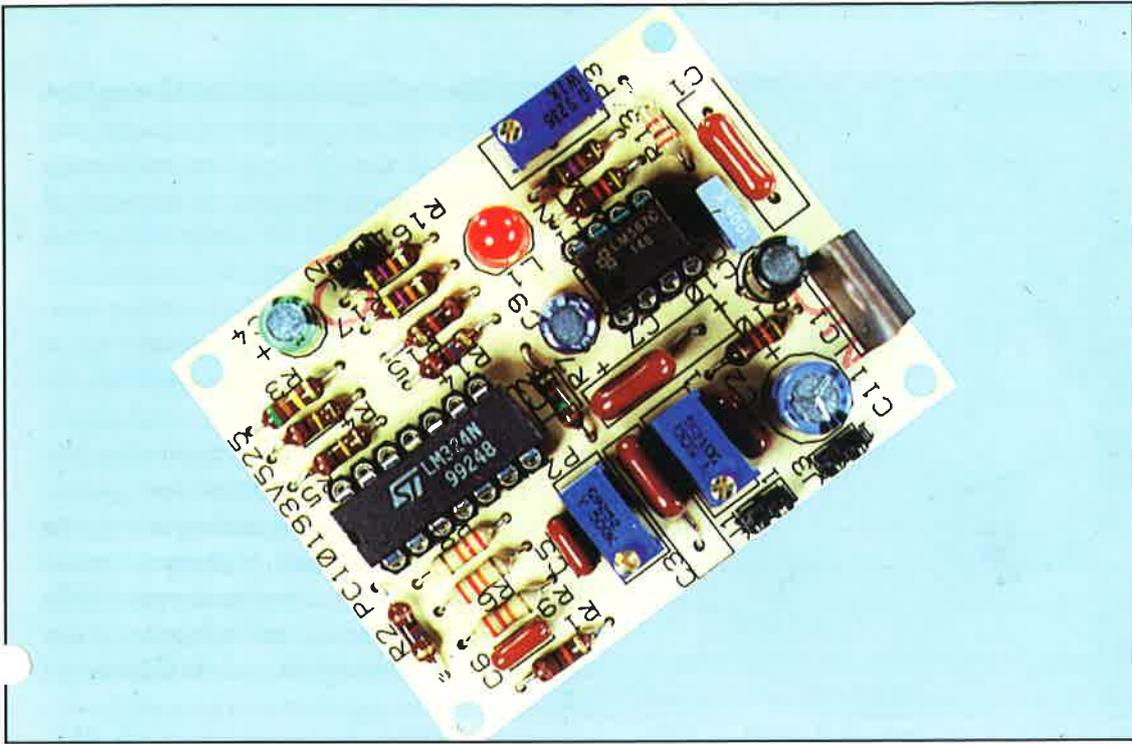
rispetto al corrispondente valore relativo alla trasmissione di tipo seriale, soprattutto se il numero dei bistabili che formano il contatore è elevato.
Tuttavia, la trasmissione in parallelo presenta alcuni inconvenienti:
- maggior numero di ingressi delle porte,
- elevato carico che deve essere sopportato dai primi bistabili del contatore.

Dopo aver descritto le due tecniche di trasmissione dei dati nei contatori sincroni, di seguito viene esaminato un tipo particolare di dispositivo molto conosciuto dai progettisti elettronici.

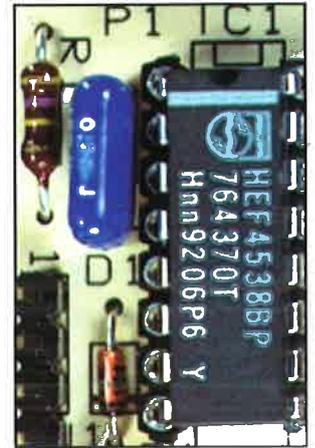
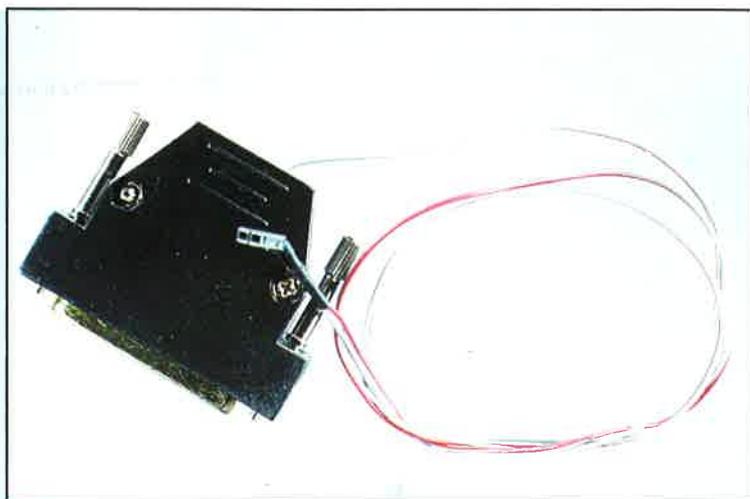
Disposizione interna di un circuito integrato che contiene un contatore sincrono up/down

CONTATORE REVERSIBILE

Questo contatore utilizza la trasmissione di tipo parallelo. Un contatore diventa reversibile (up/down) se per gli accoppiamenti tra gli stadi viene utilizzata l'uscita /Q dei bistabili invece dell'uscita Q. Partendo da questo concetto è possibile costruire un contatore sincrono reversibile semplicemente inserendo alcune porte di gestione e di controllo tra i flip-flop, come si può osservare nella figura corrispondente. Queste porte di controllo sono costituite da una porta NAND-NAND.
Se le porte di controllo di un contatore asincrono reversibile erano dotate di due ingressi, in un contatore sincrono il numero degli ingressi aumenta proporzionalmente al numero dei bistabili utilizzati. Gli ingressi aggiuntivi di queste porte sono utilizzati per la trasmissione parallela; detto in altro modo, i blocchi di controllo soddisfano entrambe le logiche, quella reversibile e quella della trasmissione in parallelo.
Esistono altri tipi di contatori, come quelli decadici, la cui funzione principale è quella della divisione per un numero multiplo di due della frequenza del segnale di clock di ingresso; anche se sono disponibili in commercio, i contatori decadici sincroni richiedono una configurazione estremamente complessa ed un costo di realizzazione molto elevato.

Circuito integrato in tecnologia CMOS che contiene un contatore binario up/down

Programma per il codificatore/ decodificatore Morse

È giunto il momento di passare alla fase operativa e di fornire il software necessario per il funzionamento dei circuiti di codifica e decodifica del codice Morse. I programmi descritti sono semplici da gestire, e possono essere facilmente modificati da parte dei lettori.

La gestione dei circuiti di interfaccia descritti nei capitoli precedenti è semplice e immediata se si utilizzano i programmi forniti con il dischetto allegato. Come si può facilmente dedurre, osservando il nome dei file, uno di questi è destinato all'interpretazione del segnale codificato che entra nella scheda di decodifica e arriva

Il codice che entra nella scheda di decodifica arriva al PC attraverso la porta Centronics

Il cavo DB-25 è indispensabile per collegare il computer alle schede di interfaccia

al PC attraverso la porta Centronics, e consente la sua visualizzazione come testo sullo schermo del computer. Il secondo consente di digitare un testo tramite la tastiera e inviarlo alla scheda di codifica, dotata di una uscita che permette la gestione dell'ingresso dell'apparecchiatura utilizzata per la trasmissione del codice Morse.
I circuiti e il software, peraltro molto semplici, richiedono alcune regolazioni che vengono descritte di seguito.

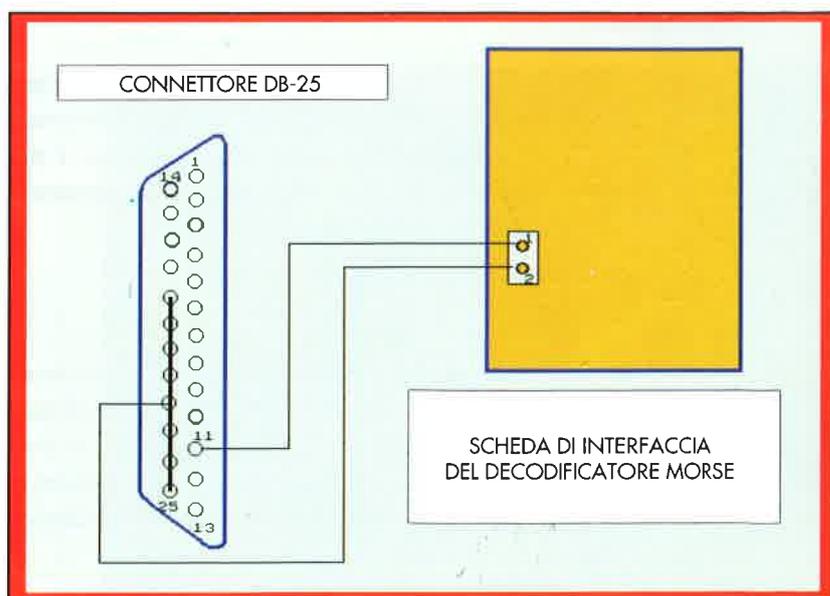
REGOLAZIONE DELLA SCHEDA DI DECODIFICA

Per prima cosa, seguendo un ordine cronologico, vengono esaminate le regolazioni necessarie per la messa a punto del decodificatore.
Come si può osservare sia sulla scheda che nella figura di riferimento, il circuito è dotato di tre potenziometri di regolazione, indicati con P1, P2 e P3. Il primo di questi (P1) serve per l'adattamento del segnale di ingresso; questa regolazione è molto importante per la successiva compensazione o attenuazione del segnale stesso da parte dall'amplificatore, in funzione del fatto che lo stesso sia più o meno elevato rispetto al livello desiderato. Questo livello è determinato dal margine di lavoro del decodificatore di toni PLL (costituito dal circuito integrato LM567). La tensione di ingresso di questo circuito deve avere un valore maggiore o uguale a 200 mV. Per eseguire la regolazione bisogna applicare un segnale Morse all'ingresso del decodificatore ed agire su P1 finché non si ottiene il valore indicato. Dopo aver verificato che il valore del segnale di ingresso è compreso nella gamma di lavoro del PLL, si può passare alla messa a punto degli altri potenziometri.

Sulla scheda sono montati tre potenziometri di regolazione indicati con P1, P2 e P3

La regolazione di P2 viene inizialmente eseguita in modo "provvisorio"; infatti, bisogna spostare il cursore di questo potenziometro al centro della sua corsa di regolazione in modo da poter operare con il filtro passa-banda in una condizione non molto critica.
Fatto ciò è possibile eseguire la regolazione di P3. Il diodo LED rappresenta un aiuto molto valido, poiché consente una regolazione visiva dell'apparecchiatura. Inizialmente si deve spostare anche il cursore di P3 al centro della sua corsa di regolazione (per ottenere una maggior precisione è consigliabile utilizzare potenziometri multigiri a regolazione verticale); successivamente bisogna eseguire la regolazione propriamente detta, che consiste in un lento spostamento del cursore di questo potenziometro finché si raggiunge il punto esatto nel quale il diodo LED inizia a illuminarsi. Questo non è di solito il punto di lavoro

Il collegamento del decodificatore deve essere eseguito come mostrato in figura

Bisogna regolare il potenziometro P2 in modo che la transizione acceso/spento del LED sia la più rapida possibile

ideale (a meno che la fortuna non sia dalla vostra parte); per poterlo ottenere è necessario eseguire ancora qualche ritocco. E' possibile che il LED si illumini in corrispondenza di un solo punto della corsa di P3, ma ciò è dovuto principalmente alla regolazione di P1 e, pertanto, al livello del segnale di ingresso; in questo caso bisogna bloccare la regolazione in quel punto. Se, al contrario, il LED si illumina in corrispondenza di un intervallo più o meno ampio della corsa del potenziometro, è necessario portare il cursore di quest'ultimo nel punto centrale di questo intervallo di attivazione.

POTENZIOMETRI DI REGOLAZIONE
DELLA SCHEDA DI DECODIFICA
DEL CODICE MORSE
Regolazione del P.L.L.
P3
P2
P1
Filtro passa-banda
Attenuazione

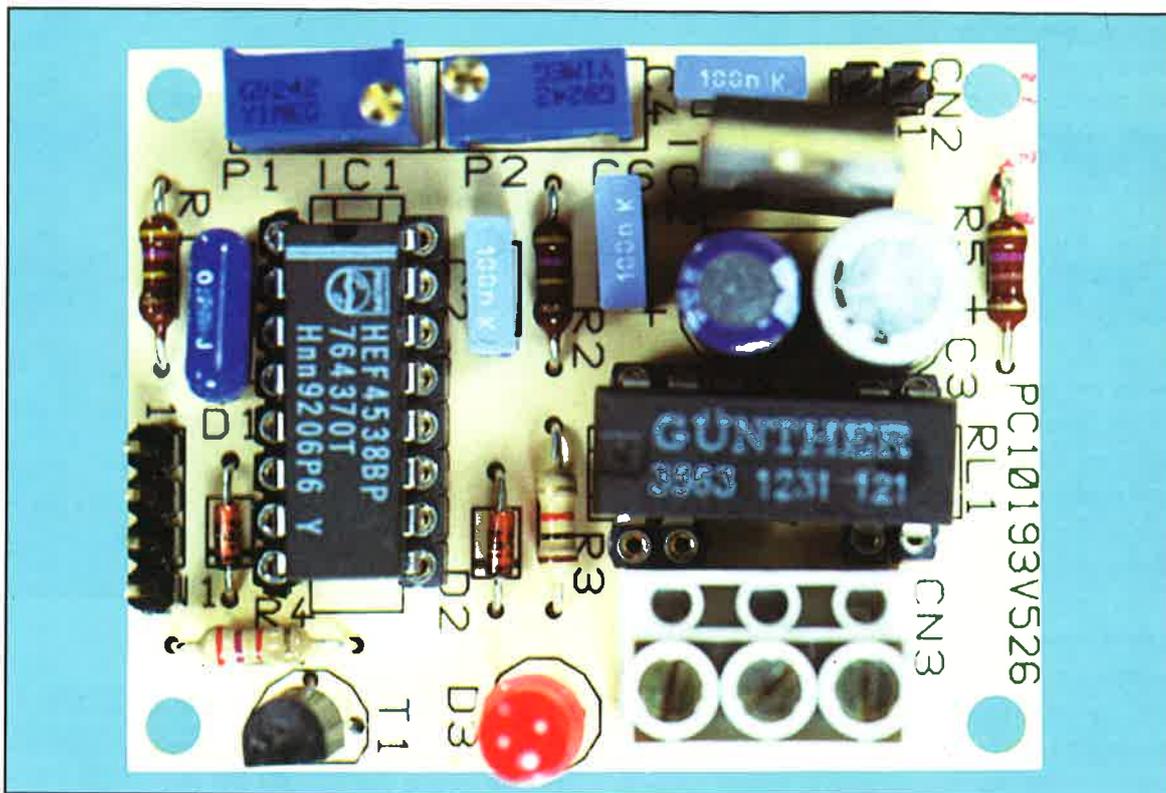
Potenziomenti per la regolazione della scheda di decodifica Morse

Supponendo che tutto quanto sia stato eseguito correttamente e senza particolari difficoltà, si può passare alla regolazione del filtro passa-banda agendo nuovamente sul potenziometro P2. Come già detto, se il segnale Morse che arriva al decodificare fa lampeggiare il LED in accordo con la frequenza degli impulsi che formano il messaggio si possono considerare le altre regolazioni come definitive, per cui è possibile operare esclusivamente con P2. Generalmente a questo punto la luce emessa dal LED, anche se condizionata dalla regolazione del segnale di ingresso (P1), deve essere sufficientemente intensa; questa condizione indica che il livello del segnale Morse è ottimale e non richiede ulteriori regolazioni.

Se si prova a spostare il cursore di P2 verso uno qualsiasi dei suoi estremi, si può notare una certa variazione (poco percettibile) nel ritmo del lampeggiamento del diodo LED; per ottimizzare la ricezione è necessario posizionare il cursore del potenziometro nel punto in cui la transizione da acceso a spento del LED avviene più rapidamente. Se la regolazione di P2 non viene eseguita in modo adeguato, si produrranno dei falsi inneschi del PLL, con una conseguente decodifica non corretta del segnale ricevuto.

Schema di collegamento tra il DB-25 e la scheda di codifica

Per ottenere una regolazione ottimale dell'interfaccia di decodifica potrebbe essere necessario ripetere alcune delle fasi suddette più di una volta; al termine però, il lampeggio del LED deve essere perfettamente sincronizzato con gli impulsi del segnale di ingresso, ed avere una luminosità accettabile senza falsi impulsi dovuti a linee o punti non reali. Con un minimo di pazienza è possibile ottenere una

Entrando nel programma di controllo, l'indicatore di stato commuta nella posizione di "attesa"

Aspetto della scheda di codifica completamente montata

regolazione visiva corretta della scheda.
La prova del fuoco però, vale a dire la verifica strumentale della funzionalità del dispositivo, si può ottenere solamente eseguendo il programma per la gestione del decodificatore fornito con il dischetto allegato.

PROGRAMMA DI CONTROLLO DEL DECODIFICATORE

Per caricare il programma di controllo della scheda di decodifica (è possibile avere altre informazioni sul software fornito con questo fascicolo leggendo il file LEGGIMI.DOC presente sul dischetto allegato) bisogna digitare il suo nome sulla linea di comando del DOS del calcolatore: sullo schermo compare la videata iniziale del programma.
In questa videata si può osservare (oltre al logotipo superiore) la presenza di una zona centrale di colore bianco, corrispondente alla zona riservata per la visualizzazione del testo decodificato e per i messaggi generati dal programma stesso. Nella parte inferiore sinistra sono disponibili quattro tasti funzione preprogrammati, la cui azione verrà esaminata di seguito; infine, nella zona inferiore destra sono presenti gli indicatori di configurazione, che informano l'utente sulle condizioni operative del programma.
Avviando il programma, l'indicatore di stato si pone in condizione di "attesa" finché non arriva un segnale all'interfaccia e non viene attivato il processo di decodifica. Quest'ultima operazione può essere eseguita in due diversi modi: premendo F1 o premendo F2. Inizialmente si consiglia di utilizzare l'opzione F1, che essendo leggermente più lenta rispetto a quella relativa ad F2 consente una regolazione più precisa del decodificatore.
Come è già stato detto in precedenza, la trasmissione Morse è, per definizione e per la normativa dei tempi a cui fa riferimento, totalmente relativa. Ciò significa che il computer non dispone di un algoritmo "fisso" per l'interpretazione del codice che riceve. Per questa ragione il programma deve autoregolarsi in funzione del tipo di codice ricevuto (durata e spaziatura dei simboli); questa funzione viene svolta in modo automatico premendo uno dei due tasti di regolazione suddetti.
Inizialmente la funzione F2 potrebbe non essere sufficientemente precisa; potrebbe però diventare utile nel momento in cui si verifica un qualsiasi inconveniente nella trasmissione (una interruzione

momentanea della ricezione, un cambio di operatore, ecc.). In questo caso è possibile utilizzare la regolazione rapida poiché, anche se meno precisa (il numero delle letture oggetto della campionatura è inferiore), potrebbe comunque essere sufficiente per riattivare la comunicazione in modo veloce.
Si può notare inoltre, che il tasto F4 è del tipo a *commutazione* (=toggle), e consente di tornare al menu iniziale quando la decodifica è già iniziata. Questa circostanza può capitare, ad esempio, quando si interrompe l'esecuzione per visualizzare il testo già tradotto, riprendendo in seguito il processo di decodifica. A questo punto si possono verificare due situazioni: nel primo caso potrebbe succedere che la stazione trasmittente (o l'operatore) non sia più la stessa, per cui è consigliabile reinizializzare il processo utilizzando i tasti F1 o F2 per reimpostare nuovamente i parametri interni dell'algoritmo di decodifica, ottenendo una maggior precisione nella decodifica successiva. La seconda ipotesi prevede che la decodifica si sia interrotta solo sullo schermo (ritorno al menu principale), e che non siano cambiati né la stazione trasmittente né l'operatore; in questo caso, se si desidera riprendere il processo, si può utilizzare il tasto F3 e passare direttamente alla decodifica, in quanto non sono necessarie ulteriori regolazioni dei parametri dell'algoritmo. É opportuno ricordare che il tasto F3 diventa attivo solo se in precedenza è stata rilevata la ricezione di un segnale Morse.
Di seguito vengono brevemente esaminati i messaggi riportati nella sezione di CONFIGURAZIONE e generati dal programma.

POTENZIOMETRI DI REGOLAZIONE
DELLA SCHEDA DI CODIFICA
DEL CODICE MORSE

Lunghezza della linea
Lunghezza del punto
P1
P2

I potenziometri P1 e P2 consentono di regolare la lunghezza della linea e del punto

I messaggi relativi all'indicatore di *Segnale* forniscono informazioni sullo stato dello stesso: se si sta ricevendo un segnale (Presente), se questo non è presente (Assente), se questo è stato rilevato correttamente (Rilevato), o se è stata tentata la rilevazione con esito negativo (Segnale non presente). L'indicatore di *Stato* fornisce invece informazioni sulla condizione in cui si trova in ogni istante il programma: *Decodifica, Attivata la regolazione dei parametri dei dati in ingresso,* oppure *Rilevata* la presenza o meno del segnale Morse. Con un minimo di pratica non ci saranno problemi per la gestione del programma; inoltre,

L'indicatore etichettato con "Segnale" fornisce notizie relative alla ricezione dello stesso

Il programma di codifica imposta la regolazione in funzione della velocità del computer

Il programma richiede successivamente di digitare il testo che si desidera trasmettere in codice Morse

è possibile modificarlo e configurarlo a proprio piacimento e, perché no, anche migliorarlo. Infine, è opportuno segnalare che questo programma è piuttosto lento, per cui il suo utilizzo non è consigliato su elaboratori di tipo XT oppure AT 286 con frequenza di lavoro inferiore ai 12 MHz.

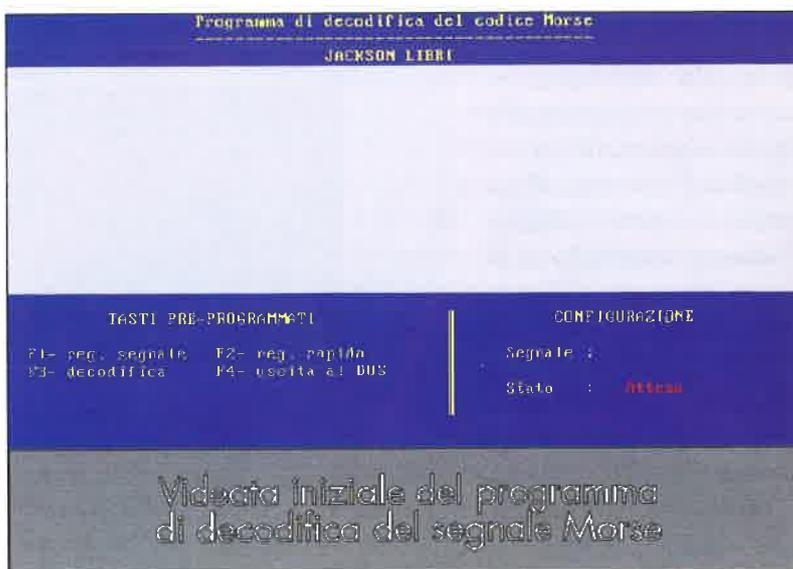
REGOLAZIONE DEL CODIFICATORE

Si verificherà che la regolazione del codificatore è molto meno impegnativa della regolazione del decodificatore. Per eseguirla sono disponibili due potenziometri: P1 e P2. Il primo di questi (P1) consente un controllo "hardware" della durata della linea, mentre con il secondo si esegue la stessa operazione sulla durata del punto.
É consigliabile in un primo momento posizionare i cursori dei potenziometri al centro della loro corsa, per poterli regolare facilmente in seguito, ottenendo i tempi che appariranno più adeguati. Il programma è molto utile per poter eseguire questa operazione, per cui non resta che esaminarne il funzionamento.

Si consiglia di impostare inizialmente entrambi i potenziometri al centro della loro corsa di regolazione

IL PROGRAMMA DI CONTROLLO DELLA SCHEDA DI CODIFICA

Questa routine, sicuramente più semplice rispetto alla precedente, risulta facile da comprendere e da modificare da parte del lettore, che dovrà semplicemente avere a disposizione un compilatore QuickBasic (o similare), oppure un interprete per questo linguaggio di programmazione.
L'invio di dati scritti sullo schermo al circuito di interfaccia sfrutta una subroutine di codifica che assegna a ciascun carattere la combinazione dei punti e delle linee che costituiscono il codice Morse. Il tempo di durata di questi può essere regolato, come detto in precedenza, tramite P1 e P2. I problemi cominciano quando gli impulsi inviati sono talmente veloci da provocare un innesco continuo del corrispondente monostabile, oppure quando si cambia il sistema informatico (con un ritmo o clock diverso dal precedente).
Per evitare queste situazioni si è complementata una lettura (realizzata con il comando INP) dello stato nel quale si trova l'emissione del simbolo (punto o linea) corrispondente, rallentando la spaziatura sia hardware che software tra gli stessi.
La gestione del programma non presenta alcuna

Videata iniziale del programma per la decodifica del codice Morse

difficoltà, ed è disponibile una utility contenuta nel programma stesso che serve per eseguire la regolazione del tempo di durata dei simboli Morse.

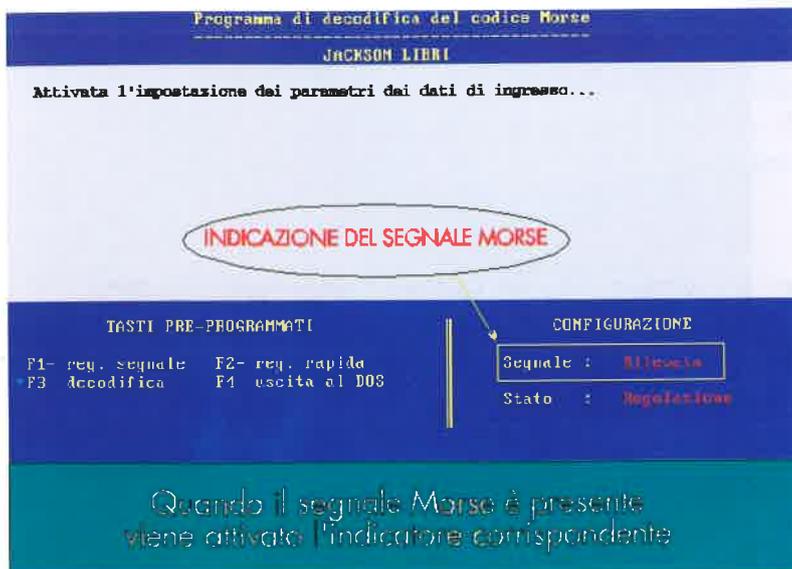
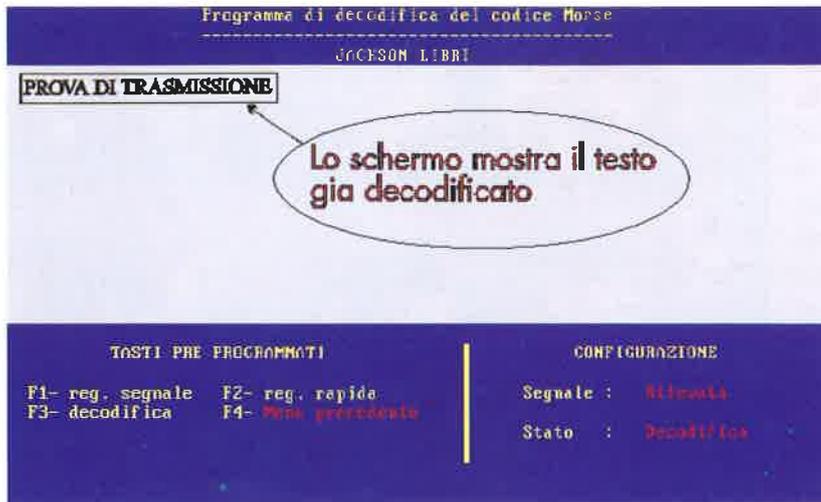
Come è già stato detto più volte, la normativa Morse tende a rendere costante la durata del rapporto punto/linea. Sempre a causa della relatività intrinseca di questo tipo di trasmissione, è regola accettata che la linea debba avere una durata pari (approssimativamente) a quella di tre punti, e quindi con un rapporto 1:3.

Per eseguire la corretta regolazione di P1 e P2, in modo che questo rapporto rimanga costante, si dovrebbe essere degli autentici maghi o possedere un udito straordinariamente sensibile. Nel progetto della scheda questo rapporto è stato previsto e configurato (C1 ha un valore pari a tre volte quello di C2), ma si è comunque voluta fornire una possibilità di regolazione fine tramite l'inserimento dei potenziometri P1 e P2 al posto di due resistenze di valore fisso.

La prima routine del programma può risultare tanto didattica quanto utile. Questa esegue una campionatura dello stato di preregolazione dei due monostabili, restituendo al programma un valore (espresso in numero di letture) proporzionale al tempo di durata dell'uscita attiva di questi circuiti: in altre parole la durata del punto e della linea. Poiché ogni elaboratore contiene una CPU che opera a velocità diversa, può capitare che alcuni di questi interpretino un segnale come una linea mentre altri, con frequenza di clock interna superiore, interpretino lo stesso segnale come un punto.

Il terminale 3 del connettore CN1 ha il compito di fornire un segnale di reazione al computer per indicare inequivocabilmente che la trasmissione del dato inviato è stata eseguita. La routine che gestisce questo segnale può essere utilizzata a proprio vantaggio per realizzare una regolazione fine della temporizzazione. L'operazione è molto semplice: infatti, quando si avvia il programma compaiono sullo schermo una coppia di valori che corrispondono rispettivamente al numero di letture realizzate dal ciclo corrispondente durante il periodo nel quale è attivato il punto o la linea.

Due schede e un connettore DB-25 è tutto ciò che serve per convertire il PC in un "radioamatore"

Prima di iniziare le sue autoregolazioni, il programma indica sullo schermo la presenza del segnale

Per inviare i dati "scritti" sullo schermo verso il circuito di interfaccia viene utilizzata una subroutine di codifica

Il testo che si sta ricevendo viene visualizzato nella parte di schermo riservata a questa funzione. In questo caso la parola "PROVA DI TRASMISSIONE"

Dal terminale 3 del connettore CN1 viene inviato al computer un segnale che indica inequivocabilmente che la trasmissione del dato è terminata

Come si può facilmente capire queste letture non possono essere interpretate in valore assoluto, poiché apparecchiature diverse forniscono letture diverse. Quello che si può (e si deve) fare è prendere nota della differenza relativa tra le letture di entrambe le campionature, per poterla reimpostare nel caso venisse modificata la posizione del potenziometro corrispondente (P1 per la linea e P2 per il punto), ottenendo in questo modo una regolazione praticamente perfetta della differenza relativa tra la dùrata del punto e quella della linea. Il funzionamento del programma operativo è più evidente, e pertanto diminuisce l'importanza di un suo esame approfondito. Infatti prosegue richiedendo l'immissione del testo che si desidera trasmettere (che viene gestito come una semplice sequenza di caratteri), lo scompone lettera per lettera, e lo invia in questa forma alla soubroutine di assegnazione del codice corrispondente alla lettera. Dopo che è stato ricavato il codice corrispondente, il programma lo invia ai primi due bit dati del bus associato alla porta parallela. Il circuito di codifica si incarica di fare il resto.

CONSIDERAZIONI FINALI

Come si può osservare nei listati sorgenti scritti in linguaggio Basic, le subroutine di assegnazione lettera/codice e codice/lettera di entrambi programmi sono in qualche modo simili. Entrambe utilizzano una codificazione propria che converte i punti in "uno" e le linee in "due", e viceversa. Se il lettore lo desidera, è possibile modificare e migliorare queste routine grazie alla semplicità delle stesse che può servire come base per ulteriori sviluppi software, sia per aumentare la velocità (auspicabile nel caso del decodificatore) sia per migliorare le prestazioni che questi circuiti possono fornire.
Si ricorda ancora la limitazione dovuta alla velocità di gestione del programma di decodifica. Per ottenere risultati soddisfacenti è consigliabile l'impiego, come configurazione minima, di un computer 286 a 12 MHz.

Videata iniziale del programma di codifica, che indica l'esecuzione del processo di autoregolazione